



IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re the Application of:

DONG-KWAN KIM, ET AL.

For: **APPARATUS AND METHOD FOR
GENERATING ENTERPRISE JAVA BEANS
BASED ON CLASS DIAGRAM**

Honorable Commissioner of
Patents and Trademarks
Washington, D.C. 20231

Request for Priority

Sir:

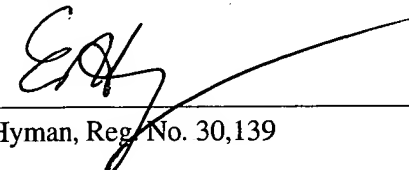
Applicant respectfully requests a convention priority for the above-captioned application, namely Korean application number 2001-1943 filed January 12, 2001.

☒ A certified copy of the document is being submitted herewith.

Respectfully submitted,

BLAKELY, SOKOLOFF, TAYLOR & ZAFMAN

Dated: 8/7/01


Eric S. Hyman, Reg. No. 30,139

12400 Wilshire Blvd., 7th Floor
Los Angeles, California 90025
Telephone: (310) 207-3800

<Priority Document Translation>



THE KOREAN INDUSTRIAL
PROPERTY OFFICE

This is to certify that the following application annexed
hereto is a true copy from the records of the Korean Industrial
Property Office.

Application Number : 2001-1943 (Patent)

Date of Application : January 12, 2001

Applicant(s) : ELECTRONICS AND TELECOMMUNICATIONS
RESEARCH INSTITUTE

April 4, 2001

COMMISSIONER



별첨 사본은 아래 출원의 원본과 동일함을 증명함.

This is to certify that the following application annexed hereto is a true copy from the records of the Korean Intellectual Property Office.

출원 번호 : 특허출원 2001년 제 1943 호
Application Number

출원 년 월 일 : 2001년 01월 12일
Date of Application

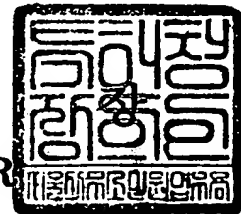
출원인 : 한국전자통신연구원
Applicant(s)



2001 년 04 월 04 일

특 허 청

COMMISSIONER



【서류명】	특허출원서
【권리구분】	특허
【수신처】	특허청장
【제출일자】	2001.01.12
【발명의 명칭】	클래스 다이어그램을 기반으로 한 엔터프라이즈 자바빈 생성 장치 및 그 방법
【발명의 영문명칭】	An Apparatus and Method for Generating Enterprise Java Beans based a Class Diagram
【출원인】	
【명칭】	한국전자통신연구원
【출원인코드】	3-1998-007763-8
【대리인】	
【성명】	특허법인 신성 정지원
【대리인코드】	9-2000-000292-3
【포괄위임등록번호】	2000-051975-8
【대리인】	
【성명】	특허법인 신성 원석희
【대리인코드】	9-1998-000444-1
【포괄위임등록번호】	2000-051975-8
【대리인】	
【성명】	특허법인 신성 박해천
【대리인코드】	9-1998-000223-4
【포괄위임등록번호】	2000-051975-8
【발명자】	
【성명의 국문표기】	김동관
【성명의 영문표기】	KIM,Dong Kwan
【주민등록번호】	700817-1068225
【우편번호】	157-040
【주소】	서울특별시 강서구 염창동 우성아파트 302-908
【국적】	KR
【발명자】	
【성명의 국문표기】	양영종
【성명의 영문표기】	YANG,Young Jong
【주민등록번호】	560308-1010433

【우편번호】	305-333
【주소】	대전광역시 유성구 어은동 한빛아파트 117-902
【국적】	KR
【발명자】	
【성명의 국문표기】	정효택
【성명의 영문표기】	JUNG, Hyo Taeg
【주민등록번호】	620925-1691310
【우편번호】	305-335
【주소】	대전광역시 유성구 궁동 다솔아파트 103-604
【국적】	KR
【심사청구】	청구
【취지】	특허법 제42조의 규정에 의한 출원, 특허법 제60조의 규정에 의한 출원심사를 청구합니다. 대리인 특허법인 신성 정지원 (인) 대리인 특허법인 신성 원석희 (인) 대리인 특허법인 신성 박해천 (인)
【수수료】	
【기본출원료】	20 면 29,000 원
【가산출원료】	17 면 17,000 원
【우선권주장료】	0 건 0 원
【심사청구료】	9 항 397,000 원
【합계】	443,000 원
【감면사유】	정부출연연구기관
【감면후 수수료】	221,500 원
【첨부서류】	1. 요약서·명세서(도면)_1통

【요약서】**【요약】**

본 발명은 클래스 다이어그램을 기반으로 한 엔터프라이즈 자바빈 생성 장치 및 그 방법에 관한 것으로, 컴포넌트 기반 소프트웨어 개발의 설계와 구현을 더욱 자연스럽게 하고 수작업에서 발생할 수 있는 오류를 최소화하여, 궁극적으로 소프트웨어 개발 기간 단축 및 개발 비용을 절감하기 위한 클래스 다이어그램을 기반으로 한 엔터프라이즈 자바빈 생성 장치 및 그 방법과 상기 방법을 실현시키기 위한 프로그램을 기록한 컴퓨터로 읽을 수 있는 기록매체를 제공하기 위하여, 클래스 다이어그램을 입력받기 위한 클래스 다이어그램 입력 수단; 상기 클래스 다이어그램 입력 수단을 통해 입력된 클래스 다이어그램의 클래스간 상속관계를 없애기 위한 상속관계 처리 수단; 상기 상속관계 처리 수단에서 상속관계가 제거된 클래스 다이어그램을 기반으로 엔터프라이즈 자바빈을 추출하기 위한 엔터프라이즈 자바빈 추출 수단; 상기 엔터프라이즈 자바빈 추출 수단에서 추출된 엔터프라이즈 자바빈의 리모트 인터페이스(remote interface)와 홈 인터페이스(home interface)를 생성하기 위한 엔터프라이즈 자바빈 인터페이스 생성 수단; 상기 엔터프라이즈 자바빈 인터페이스 생성 수단에서 인터페이스가 생성된 엔터프라이즈 자바빈을 상호 관련된 빈끼리 그룹핑하기 위한 엔터프라이즈 자바빈 그룹핑 처리 수단; 및 상기 엔터프라이즈 자바빈 그룹핑 처리 수단에서 그룹핑된 엔터프라이즈 자바빈에 패사드 패턴(facade pattern)을 적용하여 외부 인터페이스를 단일화시키기 위한 패사드 패턴 적용 수단을 포함하며, 컴퓨터 시스템 등에 이용됨.

【대표도】

도 3

【색인어】

클래스 다이어그램, 상속관계, 엔터프라이즈 자바빈(EJB), 엔터티 빈, 세션 빈, 그룹핑

【명세서】**【발명의 명칭】**

클래스 다이어그램을 기반으로 한 엔터프라이즈 자바빈 생성 장치 및 그 방법{An Apparatus and Method for Generating Enterprise Java Beans based a Class Diagram}

【도면의 간단한 설명】

도 1 은 종래의 엔터프라이즈 자바빈 기반의 소프트웨어 개발 과정에 대한 일예시도.

도 2 는 본 발명에 따른 엔터프라이즈 자바빈 생성 장치를 이용한 소프트웨어 개발 과정에 대한 일실시에 설명도.

도 3 은 본 발명에 따른 엔터프라이즈 자바빈 생성 장치의 일실시에 구성도.

도 4 는 본 발명에 따른 엔터프라이즈 자바빈 생성 방법에 대한 일실시에 흐름도.

도 5 는 본 발명에 따른 클래스 그룹화 과정에 대한 일실시에 흐름도.

도 6a 내지 6c 는 클래스 관계성에 대한 일실시에 설명도.

도 7 은 본 발명에 따른 클래스 그룹화 과정에 대한 일실시에 설명도.

도 8 은 본 발명에 따른 클래스 그룹화를 수행한 병원 클래스 다이어그램에 대한 일실시에 설명도.

도 9 는 본 발명에 따른 엔터프라이즈 자바빈 추출 과정에 대한 일실시에 흐름도.

도 10 은 본 발명에 따른 엔터프라이즈 자바빈을 위한 인터페이스 추가 과정에 대한 일실시에 흐름도.

도 11 은 본 발명에 따른 엔터프라이즈 자바빈을 위한 인터페이스 추가 과정에 대한 일실시에 설명도.

도 12 는 본 발명에 따른 패사드 패턴을 적용하는 과정에 대한 일실시에 설명도.

* 도면의 주요 부분에 대한 부호의 설명

301 : 클래스 다이어그램 입력부 302 : 상속관계 처리부

303 : 엔터프라이즈 자바빈 추출부

304 : 엔터프라이즈 자바빈 인터페이스 생성부

305 : 엔터프라이즈 자바빈 그룹핑 처리부

306 : 패사드 패턴 적용부

【발명의 상세한 설명】

【발명의 목적】

【발명이 속하는 기술분야 및 그 분야의 종래기술】

<19> 본 발명은 클래스 다이어그램을 기반으로 한 엔터프라이즈 자바빈 생성 장치 및 그 방법과 상기 방법을 실현시키기 위한 프로그램을 기록한 컴퓨터로 읽을 수

있는 기록매체에 관한 것으로, 더욱 상세하게는 설계 정보를 기반으로 한 코드 자동 생성을 통해 엔터프라이즈 자바빈 기반의 소프트웨어 개발 과정 중 발생할 수 있는 오류를 줄이고, 개발 기간의 단축 및 개발 비용의 절감을 위한 엔터프라이즈 자바빈 생성 장치 및 그 방법과 상기 방법을 실현시키기 위한 프로그램을 기록한 컴퓨터로 읽을 수 있는 기록매체에 관한 것이다.

<20> 네트워크의 발달로 인해 기존의 독립형 소프트웨어와는 달리 근래의 분산형 소프트웨어들은 갈수록 복잡해지고 있다. 이러한 소프트웨어 운영 환경의 복잡화와 소프트웨어 사용자들의 요구사항이 빠르게 변함에 따라 이를 충족시키기 위해 소프트웨어 개발자들은 기존에 개발된 프로그램을 재사용하는 기술에 관심을 갖기 시작했다.

<21> 소프트웨어 개발과정은 크게 분석(analysis), 설계(design), 구현(implementation), 시험 및 설치(test and installation) 과정으로 분류되며, 초기 소프트웨어 재사용 기술은 구현 과정에서 소스코드에 대한 재사용에 집중되었다.

<22> 소프트웨어 재사용 기술은 소프트웨어 개발과정과는 역으로 발전되고 있다. 다시 말해, 구현, 설계, 분석, 요구사항 정보에 대한 재사용으로 발전해 오고 있는 것이다.

<23> 설계 패턴(design pattern), 소프트웨어 아키텍처(software architecture), 프레임워크(architecture) 등의 재사용 기술을 통한 소프트웨어 재사용에 대한 연구는 수십 년간 계속되어 오고 있으며, 근래들어 컴포넌트 기반의 재사용 기술이 학계 및 업계에서 크게 대두되고 있다. 현재 크게 대두되는 컴포넌트 모델로는 자바(JAVA)를 기반으로 한 '선(SUN)사의 엔터프라이즈 자바빈(Enterprise Java Bean, EJB)'이 있다.

<24> 상기 엔터프라이즈 자바빈은 서버(server)측의 컴포넌트 모델로 외부의 인터페이스

를 통해 서비스를 제공하고, 컴포넌트 내부는 블랙박스(black-box)로 되어있다. 또한, 엔터프라이즈 빈(enterprise bean)은 컨테이너(container)라는 미들웨어(middleware)를 통해 분산 어플리케이션에 요구되는 서비스들을 제공받는다.

<25> 상기 엔터프라이즈 자바빈 관련 클래스로는 외부에 제공되는 서비스를 정의하고 있는 리모트 인터페이스(remote interface), 엔터프라이즈 빈의 생성 및 검색 기능을 제공하는 홈 인터페이스(home interface), 외부에 제공되는 서비스에 대한 구현 부분을 가지고 있는 엔터프라이즈 빈(enterprise bean)으로 구성되며 크게 엔터티 빈(entity bean)과 세션 빈(session bean)으로 구분된다.

<26> 상기 엔터티 빈은 데이터베이스 테이블의 하나의 행, 즉 레코드(record)를 나타내는 빈으로 데이터베이스를 추상화시킴으로써 빈(bean) 개발자로 하여금 좀 더 편리하게 데이터베이스 관련 작업을 수행하도록 지원한다.

<27> 상기 세션 빈은 비즈니스 로직을 수행하는 오퍼레이션을 가지는 빈(bean)으로 엔터티 빈을 통해 데이터베이스에 접근한다.

<28> 상술한 것처럼 엔터티 빈은 레코드를 나타내기 때문에 프라이머리 키(primary key)가 필요하다. 프라이머리 키는 일반적으로 별도의 클래스로 구현된다.

<29> 상기 엔터프라이즈 자바빈 컴포넌트 모델이 제공하는 트랜잭션(transaction), 영구성(persistence), 확장성 등을 이용함으로써 웹을 기반으로 한 분산 시스템을 보다 쉽게 개발할 수 있으며, 차후에 동일한 문제영역의 시스템 개발에 재사용할 수 있다.

<30> 그러나, 엔터프라이즈 자바빈을 이용한 컴포넌트 기술이 크게 각광받고 사용되고 있지만, 소프트웨어 개발자 입장에서는 해당 기술을 사용하기 위해 필수적으로 작성해야 하는

파일들이 있다. 예를 들면, 홈 인터페이스, 리모트 인터페이스, 빈 클래스들을 작성해야 한다.

- <31> 상기 해당 클래스들을 작성하는 과정에서 오류가 발생할 수 있으며, 이는 컴포넌트 기술을 사용하는 목적에 반하게 되는 문제점이 있다.
- <32> 특히, 기존 객체지향 방법론을 사용하던 개발자들은 클래스 다이어그램에서 상술한 클래스들을 추출하는 일련의 과정을 수작업으로 처리하게 됨으로써 개발의 효율성을 저해하는 문제점이 있었다.

【발명이 이루고자 하는 기술적 과제】

- <33> 본 발명은, 상기한 바와 같은 문제점을 해결하기 위하여 제안된 것으로, 컴포넌트 기반 소프트웨어 개발의 설계와 구현을 더욱 자연스럽게 하고 수작업에서 발생할 수 있는 오류를 최소화하여, 궁극적으로 소프트웨어 개발 기간 단축 및 개발 비용을 절감하기 위한 클래스 다이어그램을 기반으로 한 엔터프라이즈 자바빈 생성 장치 및 그 방법과 상기 방법을 실현시키기 위한 프로그램을 기록한 컴퓨터로 읽을 수 있는 기록매체를 제공하는 데 그 목적이 있다.

【발명의 구성 및 작용】

- <34> 상기 목적을 달성하는 본 발명의 장치는, 클래스 다이어그램을 기반으로 한 엔터프라이즈 자바빈 생성 장치에 있어서, 클래스 다이어그램을 입력받기 위한 클래스 다이어그램 입력 수단; 상기 클래스 다이어그램 입력 수단을 통해 입력된 클래스 다이어그램의

클래스간 상속관계를 없애기 위한 상속관계 처리 수단; 상기 상속관계 처리 수단에서 상속관계가 제거된 클래스 다이어그램을 기반으로 엔터프라이즈 자바빈을 추출하기 위한 엔터프라이즈 자바빈 추출 수단; 상기 엔터프라이즈 자바빈 추출 수단에서 추출된 엔터프라이즈 자바빈의 리모트 인터페이스(remote interface)와 홈 인터페이스(home interface)를 생성하기 위한 엔터프라이즈 자바빈 인터페이스 생성 수단; 상기 엔터프라이즈 자바빈 인터페이스 생성 수단에서 인터페이스가 생성된 엔터프라이즈 자바빈을 상호 관련된 빈끼리 그룹핑하기 위한 엔터프라이즈 자바빈 그룹핑 처리 수단; 및 상기 엔터프라이즈 자바빈 그룹핑 처리 수단에서 그룹핑된 엔터프라이즈 자바빈에 패사드 패턴(facade pattern)을 적용하여 외부 인터페이스를 단일화시키기 위한 패사드 패턴 적용 수단을 포함하는 것을 특징으로 한다.

<35> 한편, 본 발명의 방법은, 클래스 다이어그램을 기반으로 한 엔터프라이즈 자바빈 생성 장치에 적용되는 엔터프라이즈 자바빈 생성 방법에 있어서, 클래스 다이어그램에 존재하는 상속 관계를 제거하는 제 1 단계; 상기 상속 관계가 제거된 클래스들을 기반으로 엔터프라이즈 자바빈을 추출하는 제 2 단계; 상기 추출된 엔터프라이즈 자바빈들에게 엔터프라이즈 자바빈 인터페이스를 추가하는 제 3 단계; 상기 인터페이스가 추가된 엔터프라이즈 자바빈들을 상호 관련있는 빈들끼리 그룹화하는 제 4 단계; 및 상기 그룹화된 엔터프라이즈 빈들에게 패사드 패턴을 적용하는 제 5 단계를 포함하는 것을 특징으로 한다.

<36> 한편, 본 발명은, 대용량 프로세서를 구비한 엔터프라이즈 자바빈 생성 장치에, 클래스 다이어그램에 존재하는 상속 관계를 제거하는 제 1 기능; 상기 상속 관계가 제거된 클래스들을 기반으로 엔터프라이즈 자바빈을 추출하는 제 2 기능; 상기 추출된 엔터프라이즈 자바빈들에게 엔터프라이즈 자바빈 인터페이스를 추가하는 제 3 기능; 상기 인터페이스

가 추가된 엔터프라이즈 자바빈들을 상호 관련있는 빈들끼리 그룹화하는 제 4 기능; 및 상기 그룹화된 엔터프라이즈 빈들에게 패사드 패턴을 적용하는 제 5 기능을 실현시키기 위한 프로그램을 기록한 컴퓨터로 읽을 수 있는 기록매체를 제공한다.

<37> 상술한 목적, 특징들 및 장점은 첨부된 도면과 관련한 다음의 상세한 설명을 통하여 보다 분명해 질 것이다. 이하, 첨부된 도면을 참조하여 본 발명에 따른 바람직한 일실시예를 상세히 설명한다.

<38> 도 1 은 종래의 엔터프라이즈 자바빈 기반의 소프트웨어 개발 과정에 대한 일예시도로써, 클래스 다이어그램에서 엔터프라이즈 자바빈을 구현하는 과정을 나타낸다.

<39> 소프트웨어 개발자는 문제영역의 분석, 설계를 통해 클래스 다이어그램(101)을 산출한다.

<40> 도 1 에 도시된 바와 같이, 상기 문제영역의 분석, 설계를 통해 산출된 클래스 다이어그램(101)으로부터 엔터프라이즈 컴포넌트 모델에 따라 클라이언트(client)측에 제공되며, 비즈니스 로직(business logic)과 관련된 오퍼레이션(operation)들을 가지고 있는 리모트 인터페이스(102), 엔터프라이즈 자바빈에 대한 생성 및 파인더(finder) 관련 오퍼레이션을 정의한 홈 인터페이스(103), 비즈니스 로직에 대한 실제적인 구현 부분을 가지고 있는 엔터프라이즈 빈(104), 상기 엔터프라이즈 빈(104)을 구성하는 빈이 엔터티 빈인 경우에 프라이머리 키(105)를 구현한다.

<41> 상기 클래스 다이어그램(101)에서 컴포넌트 구현으로 가는 과정은 소프트웨어 개발자에 의해 수작업으로 이루어진다.

<42> 도 2 는 본 발명에 따른 엔터프라이즈 자바빈 생성 장치를 이용한 소프트웨어 개발 과정

에 대한 일실시에 설명도로서, 본 발명에 따른 엔터프라이즈 자바빈 생성 장치를 이용한 엔터프라이즈 자바빈 생성 과정을 개략적으로 나타낸다.

<43> 도 2 는 도 1 에서 나타낸 종래의 방식에 엔터프라이즈 자바빈 생성 장치(202)를 추가한 것이다.

<44> 종래 방식에선 수작업을 통해 이루어진 반면 도 2 에서는 엔터프라이즈 자바빈 생성 장치(202)를 통해 리모트 인터페이스(203), 홈 인터페이스(204), 엔터프라이즈 빈(205), 프라이머리 키(206) 구현을 자동화하고 있다.

<45> 이처럼 엔터프라이즈 자바빈 생성 장치(202)가 클래스 다이어그램(201)에 표현된 클래스 이름, 클래스의 속성, 클래스의 오퍼레이션, 클래스들간의 관계성(relationship) 등과 같은 정보들을 통해 엔터프라이즈 자바빈을 구현하게 된다. 물론, 클래스 다이어그램(201)과 엔터프라이즈 자바빈과의 매핑 과정에서 어느 클래스를 엔터티 빈으로 할 것인지 또는 세션 빈으로 할 것인지를 결정하게 된다.

<46> 도 3 은 본 발명에 따른 엔터프라이즈 자바빈 생성 장치의 일실시에 구성도이다.

<47> 도 3 에 도시된 바와 같이, 본 발명에 따른 엔터프라이즈 자바빈 생성 장치는 문제영역에 대한 분석, 설계를 통해 작성된 클래스 다이어그램을 입력받는 클래스 다이어그램 입력부(301), 상기 클래스 다이어그램 입력부(301)에 입력된 클래스 다이어그램의 클래스간에 존재하는 상속 관계를 제거하는 상속 관계 처리부(302), 상기 상속 관계 처리부(302)에서 상속 관계가 제거되어 상속 관계가 없는 클래스 다이어그램에서 엔터티 빈과 세션 빈을 추출하는 엔터프라이즈 자바빈 추출부(303), 상기 엔터프라이즈 자바빈 추출부(303)에서 추출된 엔터프라이즈 자바빈의 리모트 인터페이스와 홈 인터페이스를 생성

하는 엔터프라이즈 자바빈 인터페이스 생성부(304), 상기 엔터프라이즈 자바빈 인터페이스 생성부(304)에서 인터페이스가 생성된 엔터프라이즈 자바빈을 상호 관련된 빈들끼리 패키징(packaging)하는 엔터프라이즈 자바빈 그룹핑 처리부(305), 상기 엔터프라이즈 자바빈 그룹핑 처리부(305)에서 그룹핑된 엔터프라이즈 빈에 단일화된 인터페이스를 제공하기 위해 패사드 패턴을 적용하는 패사드 패턴 적용부(306)를 포함한다.

<48> 상기 엔터프라이즈 자바빈 생성 장치는 클래스 다이어그램을 입력받아 엔터프라이즈 자바빈 관련 클래스들을 생성하게 된다.

<49> 도 4 는 본 발명에 따른 엔터프라이즈 자바빈 생성 방법에 대한 일실시에 흐름도이다.

<50> 도 4 에 도시된 바와 같이, 본 발명에 따른 엔터프라이즈 자바빈 생성 방법은, 상속 관계 제거(401), 엔터프라이즈 자바빈 추출(402), 엔터프라이즈 자바빈 인터페이스 추가(403), 엔터프라이즈 자바빈 그룹화(404), 그리고 패사드 패턴 적용(405) 과정을 포함한다.

<51> 먼저, 상기 상속 관계 제거 과정(401)은 클래스 다이어그램에 존재하는 클래스들간의 상속 관계를, 집단화(aggregation) 관계를 이용한 대리(delegation) 기법으로 변경하는 과정이다. 이는, 엔터프라이즈 자바빈 컴포넌트 모델에서는 상속 관계를 효과적으로 지원하지 못하므로 이런 관계성을 대리 관계로 구현하여 상속의 효과를 얻고자 하는 것이다.

<52> 상기 엔터프라이즈 자바빈 추출 과정(402)에서는 상속 관계가 제거된 클래스들을 입력받아 엔터티 빈 또는 세션 빈을 추출하는 작업을 수행한다. 일반적으로 각각의 클래스들은 엔터티 빈이나 세션 빈으로 매핑된다. 데이터베이스에 저장되어 영속성을 유지해야 하

는 클래스는 엔터티 빈으로 매핑되는 반면, 영속성 보장이 필요없는 클래스는 세션 빈으로 매핑된다. 경우에 따라서는 하나의 클래스가 엔터티 빈과 세션 빈으로 분리되는 경우도 발생한다. 즉, 클래스의 속성부분은 엔터티 빈으로, 오퍼레이션 부분은 세션 빈으로 분리된다.

<53> 상기 엔터프라이즈 자바빈 인터페이스 추가 과정(403)에서는 상기 엔터프라이즈 자바빈 추출 과정(402)에서 추출된 각각의 빈들에게 리모트 인터페이스, 홈 인터페이스, 프라이머리 키를 생성하는 작업을 수행한다. 여기서, 상기 리모트 인터페이스는 엔터프라이즈 빈이 가지는 비즈니스 로직과 관련된 오퍼레이션을 가지고, 상기 홈 인터페이스는 엔터프라이즈 빈의 생성(create메소드)과 검색(finder메소드)에 관련된 메소드를 가진다. 또한, 상기 프라이머리 키는 엔터티 빈에만 필요하며 데이터베이스 테이블의 프라이머리 키 기능을 수행한다.

<54> 상기 엔터프라이즈 자바빈 그룹화 과정(404)에서는 엔터프라이즈 자바빈 인터페이스 추가 과정(403)에서 추출한 엔터프라이즈 빈들을 입력받아 상호 관련있는 빈들끼리 패키징한다. 이는, 상호 관련된 빈들을 그룹핑함으로써 하나의 인터페이스를 제공하여 재사용의 기본 단위가 될 수 있도록 하기 위해서이다.

<55> 상기 패사드 패턴 적용 과정(405)에서는 엔터프라이즈 자바빈 그룹화 과정(404)에서 그룹핑된 엔터프라이즈 빈들을 하나의 인터페이스로 묶어 주는 기능을 하며, 외부에서 볼 때는 하나의 통로를 통해 해당 빈들을 접근함으로써 사용이 용이하다는 장점이 있다. 여기서, 그룹핑된 빈들을 대표하는 세션 빈을 추가한다. 그리고, 해당 세션 빈은 그룹핑된 빈들이 제공하는 오퍼레이션들을 가지며, 외부에서 들어온 메시지를 받아 특정 빈으로 전달해주는 기능을 한다. 또한, 빈 사용자들은 빈 내부는 볼 수 없으며, 패사드 역할을

하는 세션 빈을 통해서 그룹핑된 빈을 사용한다.

<56> 도 5 는 본 발명에 따른 클래스 그룹화 과정에 대한 일실시에 흐름도이다.

<57> 본 발명에서는 상호 밀접한 관계를 가지는 클래스를 그룹화해서 컴포넌트 단위로 사용하며 차후에 컴포넌트 배치와 실행의 단위가 된다.

<58> 도 5 에 도시된 바와 같이, 클래스 그룹화 과정은, 클래스 다이어그램으로부터 클래스 관계성이 있는지를 판단한다(501).

<59> 상기 판단 결과, 클래스 관계성이 없으면 종료하고, 클래스 관계성이 있으면 클래스 관계성을 추출하여(502) 상속 관계인지를 판단한 후(503), 상속관계이면 상속 관계에 있는 상위 클래스와 하위 클래스를 그룹화하고(504) 다시 클래스 관계성이 있는지를 판단하는 과정(501)으로 진행하고, 상속관계가 없으면 집단화 관계인지를 판단한다(505).

<60> 상기 판단 결과(505), 집단화 관계이면 집단화 관계에 있는 전체 클래스(whole class)와 부분 클래스(part class)를 그룹화하여(506) 다시 클래스 관계성이 있는지를 판단하는 과정(501)으로 진행하고, 집단화 관계가 아니면 바로 클래스 관계성이 있는지를 판단하는 과정(501)으로 진행한다.

<61> 상기와 같이, 상속 관계도 집단화 관계도 아닌 관계성은 배제하면서 새로운 클래스 관계성이 있는지 판단하는 과정을 반복적으로 수행한다. 그리고, 상기 과정은 클래스 관계성이 없을 때까지 계속한다.

<62> 도 6a 내지 6c 는 클래스 관계성에 대한 일실시에 설명도이다.

<63> 클래스 그룹화는 클래스간의 관계성을 고려하여 그룹화하며, 상기 클래스간의 관계성은 다음과 같다.

<64> 먼저, 임의의 클래스가 다른 클래스와 관계를 가지는 두 클래스간의 관계를 말하는 연관화 관계(association relationship), 하위 클래스가 상위 클래스의 인스턴스(instance) 변수 및 메소드(method)를 마치 하위 클래스 안에서 직접 정의한 것처럼 사용하는 관계를 말하는 상속 관계(inheritance relationship), 한 클래스가 다른 클래스에서 생성된 객체들을 포함하는 관계에서 포함된 클래스의 인스턴스는 포함하고 있는 클래스의 인스턴스 변수처럼 사용되는 관계를 말하는 집단화 관계(aggregation relationship)로 분류된다.

<65> 도 6a 는 연관화 관계에 대한 일실시에 설명도로서, 선생님(teacher) 클래스(601)와 학생(student) 클래스(602)의 관계처럼 두 클래스가 상호 동등한 관계를 가지는 관계성이다.

<66> 도 6b 는 상속 관계에 대한 일실시에 설명도로서, 상위 클래스와 하위 클래스로 구성되며, 사람(person) 클래스(603)는 상위 클래스이고, 고용인(employee) 클래스(604), 고객(customer) 클래스(605), 고용주(employer) 클래스(606)는 하위 클래스이다.

<67> 도 6c 는 집단화 관계에 대한 일실시에 설명도로서, 전체 클래스와 부분 클래스로 구성되며, 스테레오(stereo) 클래스(607)는 전체 클래스이고, 수신기(receiver) 클래스(608), CD 플레이어(CD player) 클래스(609), 스피커(speaker) 클래스(610)는 부분 클래스에 해당한다.

<68> 도 7 은 본 발명에 따른 클래스 그룹화 과정에 대한 일실시에 설명도이다.

<69> 도 7 에 도시된 바와 같이, 병원 도메인에 대한 간략화된 클래스 다이어그램을 살펴보면 다음과 같다.

- <70> 먼저, 병원(hospital) 클래스(701)는 부서(department) 클래스(702)와 집단화 관계에 있으며, 간호사(nurse) 클래스(703)와 의사(doctor) 클래스(704)가 존재하며 의사(doctor) 클래스(704)에는 의사(doctor) 클래스(704)를 상속받는 응급실 의사(emergency doctor) 클래스(705)가 존재한다.
- <71> 환자(patient) 클래스(707)는 일반 환자(general patient) 클래스(710)와 응급 환자(emergency patient) 클래스(711)로 구분되며 환자(patient) 클래스(707)와 의사(doctor) 클래스(704)는 치료(treatment) 클래스(708)를 통해 관계를 가진다.
- <72> 상기 치료 클래스(708)는 처방(prescription) 클래스(709)를 부분 클래스로 가지며, 환자(patient) 클래스(707)는 치료비(payment) 클래스(706)와 연관화 관계를 가진다. 즉, 상기 클래스 다이어그램에 속하는 클래스들은 연관화, 상속, 집단화 관계를 가진다.
- <73> 도 8 은 본 발명에 따른 클래스 그룹화를 수행한 병원 클래스 다이어그램에 대한 일실시에 설명도로서, 도 7 에 나타난 병원 클래스 다이어그램에 도 6 의 클래스 그룹화를 적용한 결과를 나타낸다.
- <74> 클래스 다이어그램에서 상속 관계를 추출하면, 의사(doctor) 클래스와 응급실 의사(emergency doctor) 클래스(801) 및 환자(patient) 클래스와 일반 환자(general patient) 클래스와 응급 환자(emergency patient) 클래스(802)가 상속 관계이며, 상기 클래스들이 먼저 그룹화된다.
- <75> 상속 관계에 있는 클래스들에 대한 그룹화가 완료되면 집단화 관계에 있는 클래스들이 추출된다. 즉, 병원(hospital) 클래스와 부서(department) 클래스(803), 치료(treatment) 클래스와 처방(prescription) 클래스(804)가 집단화 관계에 있으며, 이들

클래스들이 그룹화된다.

<76> 나머지 클래스들은 연관화 관계에 있기 때문에 그룹화에서 배제된다.

<77> 도 9 는 본 발명에 따른 엔터프라이즈 자바빈 추출 과정에 대한 일실시에 흐름도이다.

<78> 상기 클래스 다이어그램으로부터 클래스를 추출하여(901) 엔터프라이즈 자바빈으로 생성할 것인지 아닌지를 판단한다(902).

<79> 상기 판단 결과, 엔터프라이즈 자바빈으로 생성하지 않으면 클래스 추출과정(901)으로 진행하고, 엔터프라이즈 자바빈으로 생성할 것이면 엔터프라이즈 자바빈으로 생성할 클래스를 추출하고(903), 상기 추출된 클래스에 대해 영속성 보장여부를 판단한다(904).

<80> 상기 영속성 보장여부를 판단한 결과, 추출된 클래스가 영속성을 가지면 영속성을 가지는 클래스를 추출하고(905), 데이터베이스에 저장될 속성 추출인지 오퍼레이션 추출인지를 판단한다(906).

<81> 상기 판단 결과, 데이터데이터베이스에 저장될 속성 추출인 경우 속성을 추출한 후(907) 엔터티 빈으로 생성한다(908).

<82> 상기 판단 결과(906), 오퍼레이션 추출인 경우 영속성을 가지는 클래스의 오퍼레이션 부분을 추출하여(909) 세션 빈으로 생성한다(910).

<83> 상기 영속성 보장여부를 판단한 결과(904), 영속성을 가지지 않는 경우 영속성을 보장하지 않아도 되는 클래스를 추출하여(911) 세션 빈으로 생성한다(910).

<84> 도 10 은 본 발명에 따른 엔터프라이즈 자바빈을 위한 인터페이스 추가 과정에 대한 일실시에 흐름도로서, 도 9 의 과정을 통해 추출된 엔터프라이즈 빈에 요구되는 인터페이스들을 추가하는 과정을 나타낸다.

- <85> 먼저, 빈 클래스를 추출하여(1001) 해당 문제영역과 관련된 비즈니스 로직 오퍼레이션을 추출하여(1002), 해당 오퍼레이션들을 리모트 인터페이스에 추가한다(1003).
- <86> 다음으로, 엔터프라이즈 빈 생성과 검색에 관련된 오퍼레이션을 홈 인터페이스에 추가한다(1004). 그리고, 특정 빈이 엔터티 빈인가를 판단하여(1005), 상기 판단 결과가 엔터티 빈이면 프라이머리 키 클래스를 추가하고(1006), 그렇지 않으면 작업을 종료한다.
- <87> 도 11 은 본 발명에 따른 엔터프라이즈 자바빈을 위한 인터페이스 추가 과정에 대한 일 실시예 설명도이다.
- <88> 상기 도 7의 병원 클래스 다이어그램에서 환자 클래스 부분을 발췌한다. 환자(patient) 클래스(1101)를 엔터프라이즈 자바빈으로 바꾸면 상속 관계를 없애고 일반 환자(general patient) 클래스(1102)와 응급 환자(emergency patient) 클래스(1103)가 추출된다.
- <89> 먼저, 일반 환자(general patient) 클래스(1102)를 빈으로 만들기 위해 일반 환자 리모트 인터페이스(GPRemoteInterface) 클래스(1107)와 일반 환자 홈 인터페이스(GPHomeInterface) 클래스(1104)가 추가되며, 일반 환자(general patient) 클래스(1102)는 엔터티 빈이므로 프라이머리 키(GPPrimaryKey) 클래스(1106)가 추가된다.
- <90> 응급 환자(emergency patient) 클래스(1103)도 일반 환자(general patient) 클래스(1102)와 마찬가지로 엔터티 빈이므로 응급 환자 리모트 인터페이스(EPRemoteInterface) 클래스(1111), 응급 환자 홈 인터페이스(EPHomeInterface) 클래스(1108), 응급 환자 프라이머리 키(EPPrimaryKey) 클래스(1110)가 추가된다.
- <91> 도 12 는 본 발명에 따른 패사드 패턴을 적용하는 과정에 대한 일 실시예 설명도이다.
- <92> 상기 패사드 패턴 적용 과정은 엔터프라이즈 자바빈 그룹화 과정에서 그룹핑된 엔터프라이즈

이즈 빈들을 하나의 인터페이스로 묶어 주는 기능을 한다.

- <93> 상기 도 11의 일반 환자 빈(GPBean) 클래스(1105)와 응급 환자 빈(EPBean) 클래스(1109)에 패사드 패턴을 적용한다.
- <94> 도면에서, 클라이언트(client) 클래스(1202)는 일반 환자 빈(GPBean)(1204)과 응급 환자 빈(EPBean)(1205)을 직접 접근할 수 없으며 단일 인터페이스인 환자관리세션빈(patient management sessionbean) 클래스(1203)를 통해서 접근할 수 있다. 즉, 상기 환자관리 세션빈(1203)은 패사드 기능을 수행하게 된다.
- <95> 상술한 바와 같은 본 발명의 방법은 프로그램으로 구현되어 컴퓨터로 읽을 수 있는 형태로 기록매체(씨디롬, 램, 롬, 플로피 디스크, 하드 디스크, 광자기 디스크 등)에 저장될 수 있다.
- <96> 이상에서 설명한 본 발명은 전술한 실시예 및 첨부된 도면에 의해 한정되는 것이 아니고, 본 발명의 기술적 사상을 벗어나지 않는 범위 내에서 여러 가지 치환, 변형 및 변경이 가능하다는 것이 본 발명이 속하는 기술분야에서 통상의 지식을 가진 자에게 있어 명백할 것이다.

【발명의 효과】

- <97> 상기한 바와 같은 본 발명은, 설계 정보에서 엔터프라이즈 자바빈 생성을 수작업으로 하거나 일부 작업에서 자동화를 시도한 종래의 기술에 비해, 본 발명에 의하여 제시된 엔터프라이즈 자바빈 생성 장치를 사용하면 엔터프라이즈 자바빈 개발에 따른 오류를 줄일

수 있으며, 결과적으로 소프트웨어 개발기간의 단축과 개발 비용을 절감할 수 있는 효과가 있다.

【특허청구범위】**【청구항 1】**

클래스 다이어그램을 기반으로 한 엔터프라이즈 자바빈 생성 장치에 있어서,

클래스 다이어그램을 입력받기 위한 클래스 다이어그램 입력 수단;

상기 클래스 다이어그램 입력 수단을 통해 입력된 클래스 다이어그램의 클래스간 상속관계를 없애기 위한 상속관계 처리 수단;

상기 상속관계 처리 수단에서 상속관계가 제거된 클래스 다이어그램을 기반으로 엔터프라이즈 자바빈을 추출하기 위한 엔터프라이즈 자바빈 추출 수단;

상기 엔터프라이즈 자바빈 추출 수단에서 추출된 엔터프라이즈 자바빈의 리모트 인터페이스(remote interface)와 홈 인터페이스(home interface)를 생성하기 위한 엔터프라이즈 자바빈 인터페이스 생성 수단;

상기 엔터프라이즈 자바빈 인터페이스 생성 수단에서 인터페이스가 생성된 엔터프라이즈 자바빈을 상호 관련된 빈끼리 그룹핑하기 위한 엔터프라이즈 자바빈 그룹핑 처리 수단; 및

상기 엔터프라이즈 자바빈 그룹핑 처리 수단에서 그룹핑된 엔터프라이즈 자바빈에 패사드 패턴(facade pattern)을 적용하여 외부 인터페이스를 단일화시키기 위한 패사드 패턴 적용 수단

을 포함하는 엔터프라이즈 자바빈 생성 장치.

【청구항 2】

제 1 항에 있어서,

상기 엔터프라이즈 자바빈 추출 수단은,

데이터베이스에 저장되어 영속성을 유지해야 하는 클래스를 엔터티 빈으로 매핑하고, 영속성 보장이 필요없는 클래스를 세션 빈으로 매핑하는 것을 특징으로 하는 엔터프라이즈 자바빈 생성 장치.

【청구항 3】

제 1 항 또는 제 2 항에 있어서,

상기 엔터프라이즈 자바빈 인터페이스 생성 수단은,

엔터티 빈이면 프라이머리 키를 생성하는 것을 특징으로 하는 엔터프라이즈 자바빈 생성 장치.

【청구항 4】

클래스 다이어그램을 기반으로 한 엔터프라이즈 자바빈 생성 장치에 적용되는 엔터프라이즈 자바빈 생성 방법에 있어서,

클래스 다이어그램에 존재하는 상속 관계를 제거하는 제 1 단계;

상기 상속 관계가 제거된 클래스들을 기반으로 엔터프라이즈 자바빈을 추출하는 제 2 단계;

상기 추출된 엔터프라이즈 자바빈들에게 엔터프라이즈 자바빈 인터페이스를 추가하는 제 3 단계;

상기 인터페이스가 추가된 엔터프라이즈 자바빈들을 상호 관련있는 빈들끼리 그룹화하는 제 4 단계; 및

상기 그룹화된 엔터프라이즈 빈들에게 패사드 패턴을 적용하는 제 5 단계를 포함하는 엔터프라이즈 자바빈 생성 방법.

【청구항 5】

제 4 항에 있어서,

상기 제 2 단계는,

상기 클래스 다이어그램으로부터 클래스를 추출하여 상기 추출된 클래스로부터 엔터프라이즈 자바빈으로 생성할 것인지를 판단하는 제 6 단계;

상기 제 6 단계의 판단 결과, 상기 엔터프라이즈 자바빈으로 생성하지 않으면 제 6 단계로 진행하고, 상기 엔터프라이즈 자바빈으로 생성하면 상기 엔터프라이즈 자바빈으로 생성할 해당 클래스를 추출하는 제 7 단계;

상기 추출된 클래스에 대해 영속성 보장 여부를 판단하는 제 8 단계;

상기 제 8 단계의 판단 결과, 영속성을 가지면 상기 영속성을 가지는 클래스를 추출하여 엔터프라이즈 빈을 생성하는 제 9 단계; 및

상기 제 8 단계의 판단 결과, 영속성을 가지지 않으면 영속성을 가지지 않는 클래스를 추출하여 세션 빈을 생성하는 제 10 단계

를 포함하는 엔터프라이즈 자바빈 생성 방법.

【청구항 6】

제 5 항에 있어서,

상기 제 9 단계는,

영속성을 가지는 클래스를 추출하는 제 11 단계;

데이터베이스에 저장될 속성 추출인지 오퍼레이션 추출인지 판단하는 제 12 단계;

상기 제 12 단계의 판단 결과, 속성 추출이면 데이터베이스에 저장될 속성을 추출하여 엔터티 빈을 생성하는 제 13 단계; 및

상기 제 12 단계의 판단 결과, 오퍼레이션 추출이면 상기 오퍼레이션을 추출하여 세션 빈을 생성하는 제 14 단계

를 포함하는 엔터프라이즈 자바빈 생성 방법.

【청구항 7】

제 4 항 내지 제 6 항 중 어느 한 항에 있어서,

상기 제 3 단계는,

추출된 빈 클래스로부터 문제영역과 관련된 비즈니스 로직 오퍼레이션을 추출하는 제 15 단계;

상기 추출된 비즈니스 로직 오퍼레이션들을 리모트 인터페이스에 추가하는 제 16 단계;

상기 엔터프라이즈 빈 생성과 검색에 관련된 오퍼레이션을 홈 인터페이스에 추가하는 제 17 단계; 및

상기 추출된 엔터프라이즈 빈이 엔터티 빈임을 확인하여 프라이머리 키 클래스를 추가하는 제 18 단계

를 포함하는 엔터프라이즈 자바빈 생성 방법.

【청구항 8】

제 7 항에 있어서,

상기 제 4 단계는,

클래스 관계성이 있는지를 판단하는 제 19 단계;

상기 제 19 단계의 판단 결과, 클래스 관계성이 있음이 확인됨에 따라 클래스 관계성을 추출하여 상속 관계인지를 판단하는 제 20 단계;

상기 제 20 단계의 판단 결과, 상속관계이면 상속 관계에 있는 상위 클래스와 하위 클래스를 그룹화하고 상기 제 19 단계로 진행하고, 상속관계가 없으면 집단화 관계인지를 판단하는 제 21 단계; 및

상기 제 21 단계의 판단 결과, 집단화 관계이면 집단화 관계에 있는 전체 클래스(whole class)와 부분 클래스(part class)를 그룹화하고 상기 제 19 단계로 진행하고, 집단화 관계가 아니면 상기 제 19 단계로 진행하는 제 22 단계

를 포함하는 엔터프라이즈 자바빈 생성 방법.

【청구항 9】

대용량 프로세서를 구비한 엔터프라이즈 자바빈 생성 장치에,

클래스 다이어그램에 존재하는 상속 관계를 제거하는 제 1 기능;

상기 상속 관계가 제거된 클래스들을 기반으로 엔터프라이즈 자바빈을 추출하는 제 2 기능;

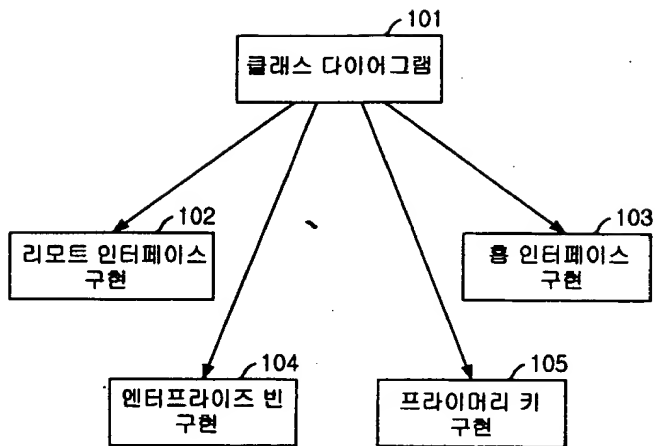
상기 추출된 엔터프라이즈 자바빈들에게 엔터프라이즈 자바빈 인터페이스를 추가하는 제 3 기능;

상기 인터페이스가 추가된 엔터프라이즈 자바빈들을 상호 관련있는 빈들끼리 그룹화하는 제 4 기능; 및

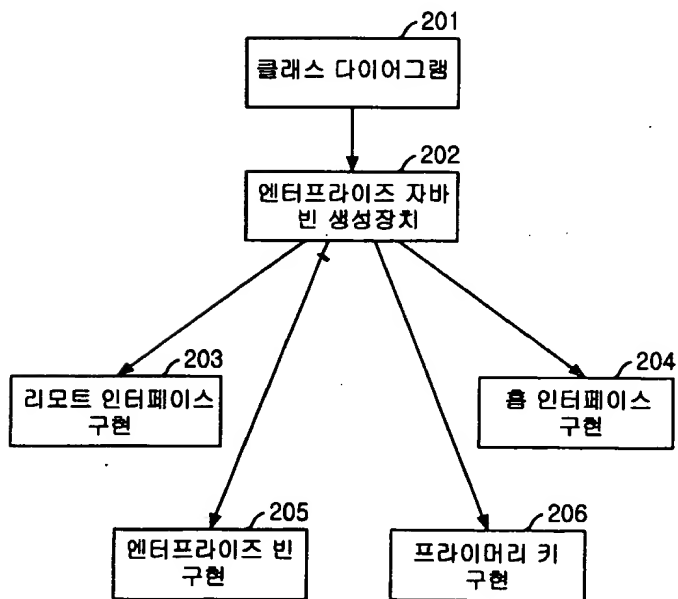
상기 그룹화된 엔터프라이즈 빈들에게 패사드 패턴을 적용하는 제 5 기능을 실현시키기 위한 프로그램을 기록한 컴퓨터로 읽을 수 있는 기록매체.

【도면】

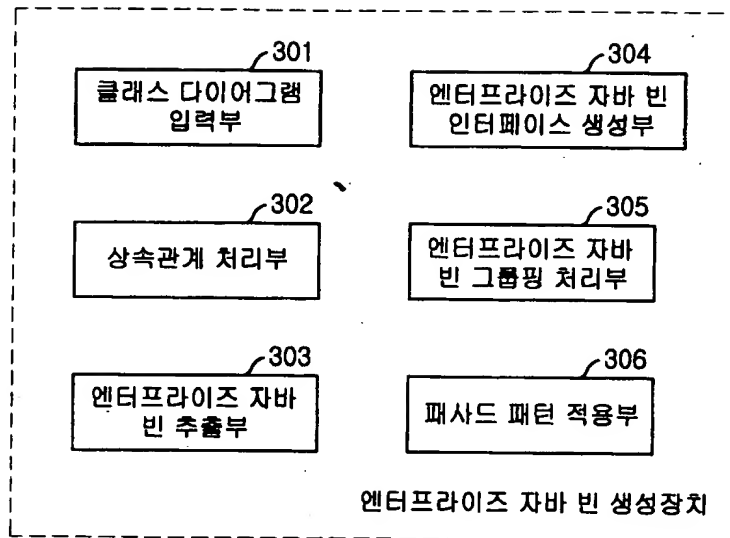
【도 1】



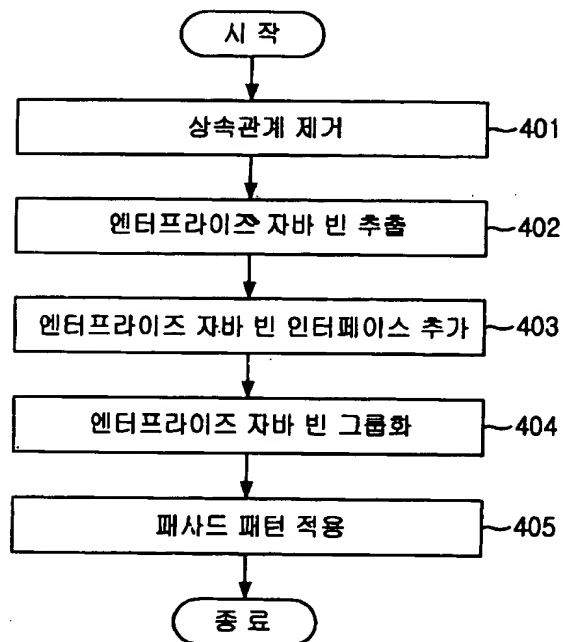
【도 2】



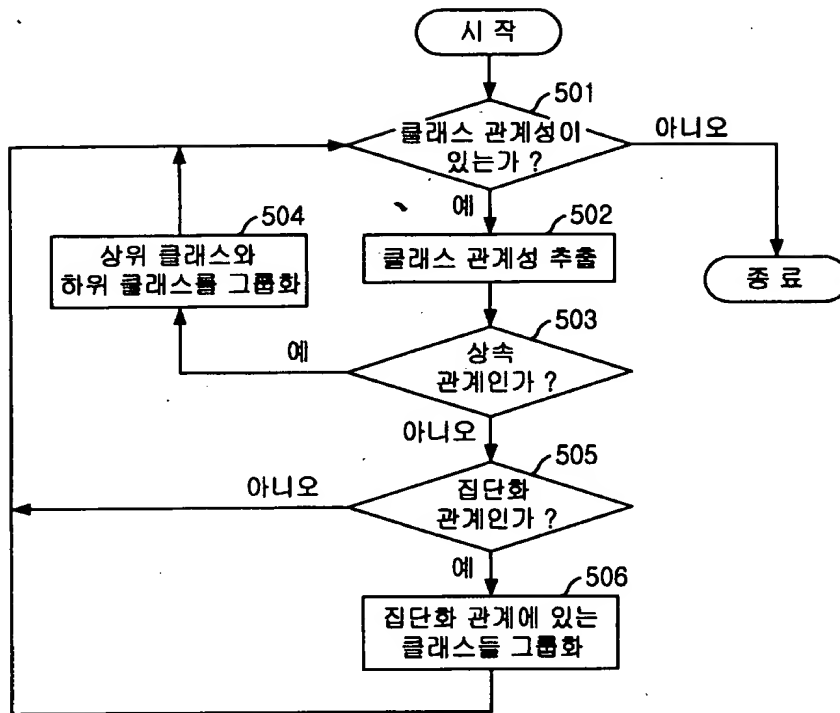
【도 3】



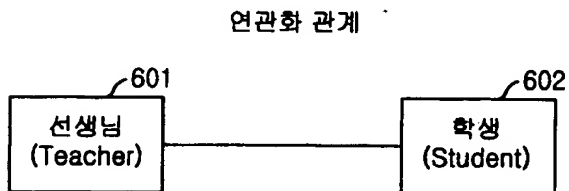
【도 4】



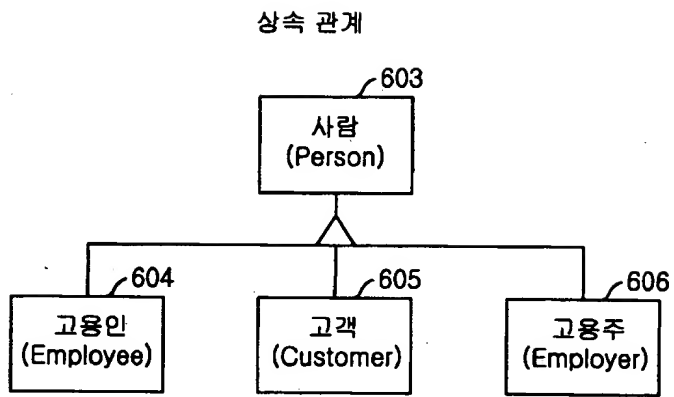
【도 5】



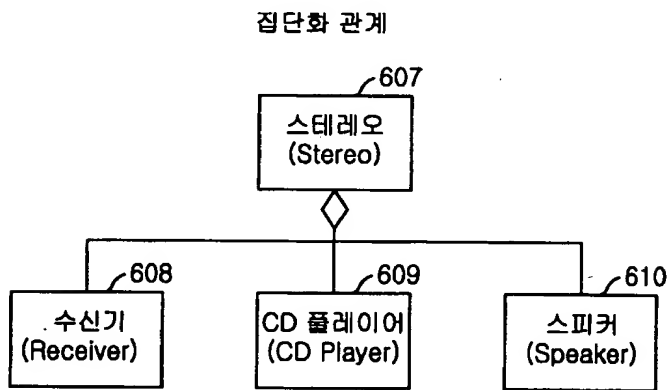
【도 6a】



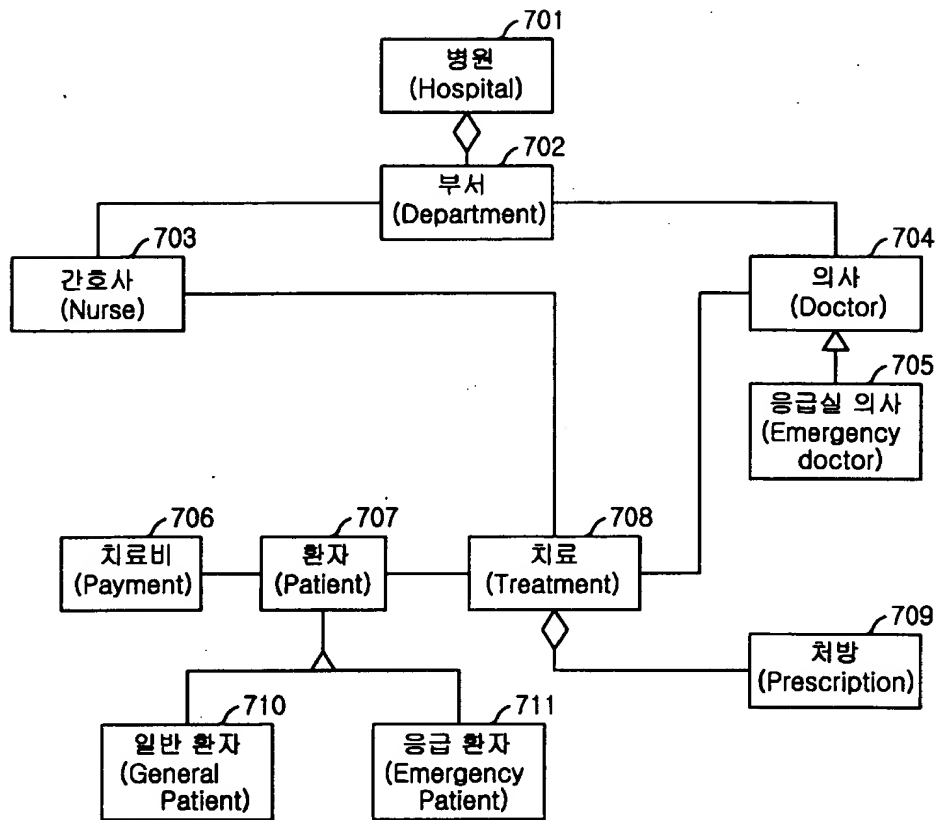
【도 6b】



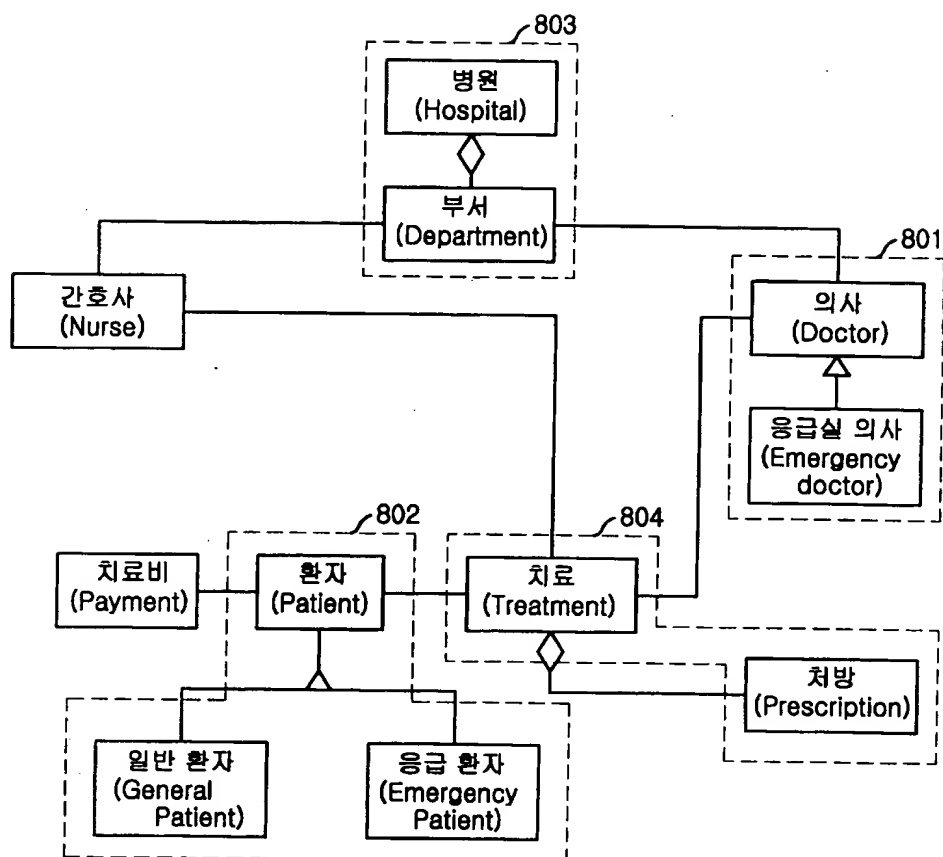
【도 6c】



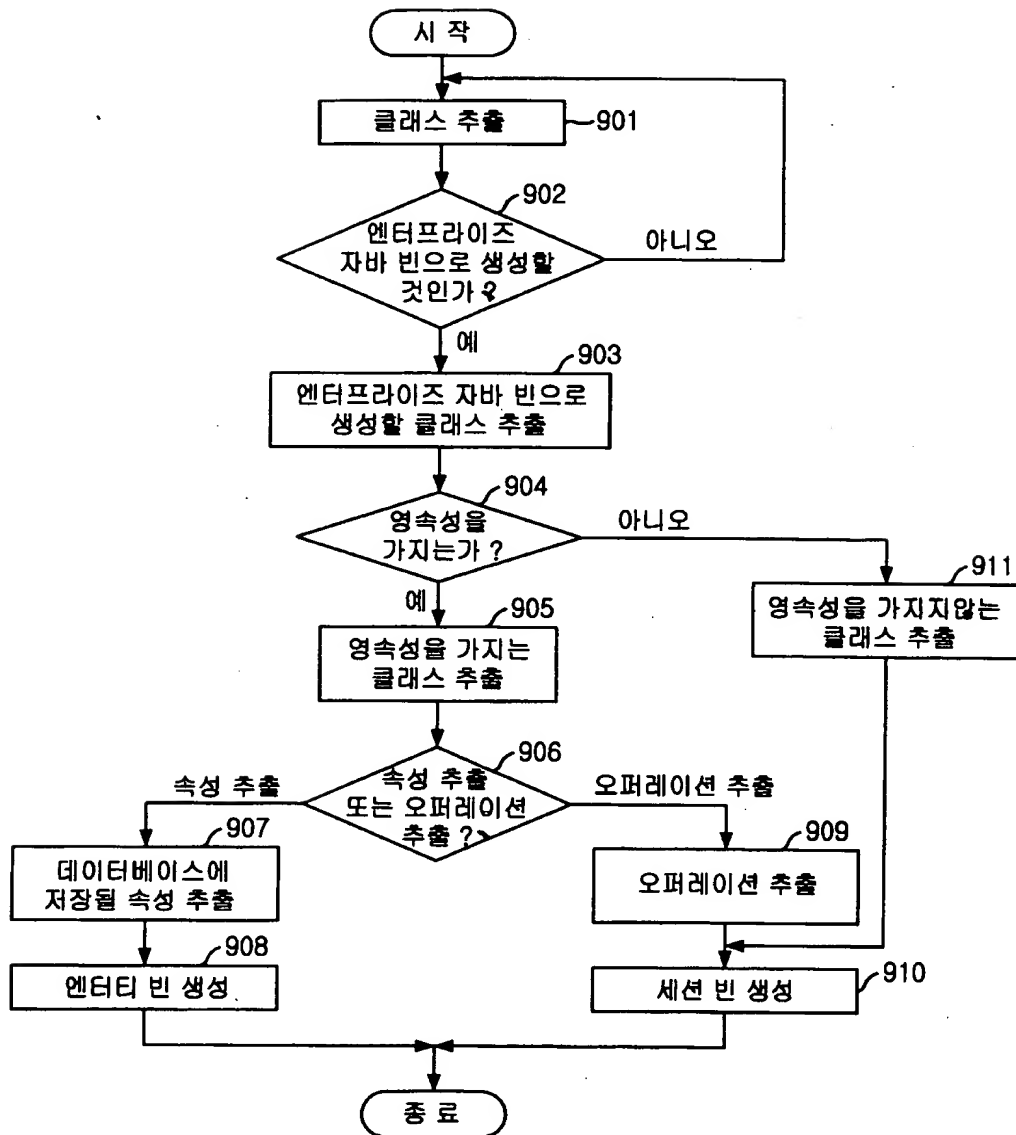
【도 7】



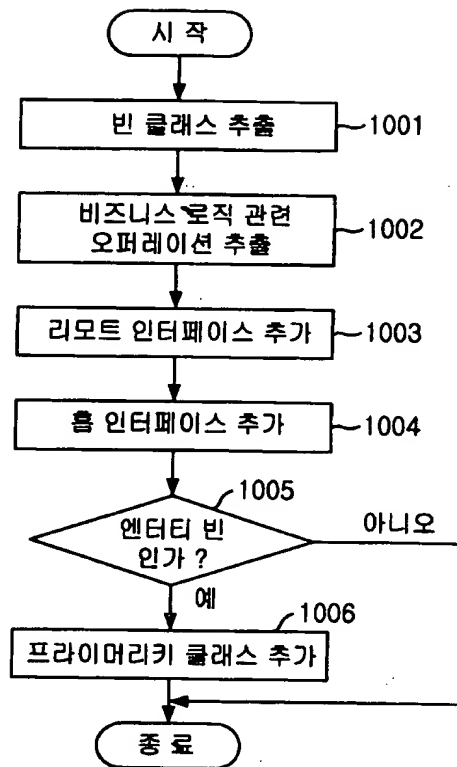
【도 8】



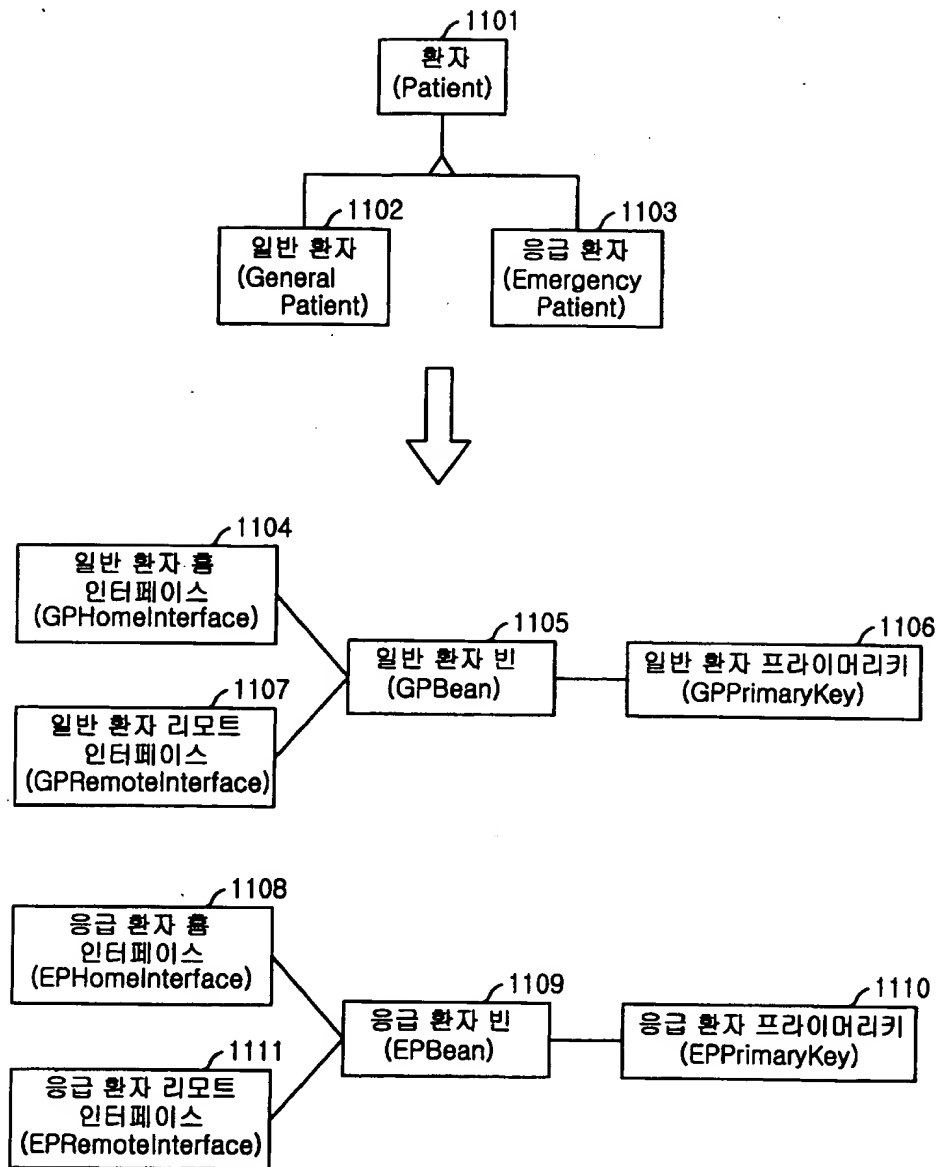
【도 9】



【도 10】



【도 11】



【도 12】

